A Hybrid PSO-BP Optimized Fuzzy Neural Network for Network Security Situation Awareness

Keguang Yang
College of Information Engineering, College of Arts and Sciences·Kunming, Kunming, 650222, China
E-mail: kmykg163@163.com

Keywords: network security situation awareness, PSO-BP algorithm, principal component analysis, fuzzy evaluation method, types of network attacks

Received: August 26, 2025

With the continuous maturity of computer technology, computers are constantly subjected to network attacks. To better respond to different types of computer network security attacks, a computer network situational awareness model based on particle swarm optimization-error back propagation algorithm is proposed. These two algorithms are optimized by introducing compression factors and momentum. Meanwhile, the model adopts a hybrid update strategy, randomly selecting the PSO or BP algorithm for parameter update at each step. In the experimental results, when the number of iterations of the proposed PSO-BP model exceeded 30, the loss function value, accuracy, recall, and F1-score converged to around 0.01, 0.99, 0.94, and 0.98, respectively. Each performance index is superior to the other two PSO-BP algorithms. In addition, compared with other models such as adaptive fuzzy system, reinforcement learning-based method, CNN, LSTM, and Transformer, the PSO-BP model demonstrates higher detection accuracy and adaptability when dealing with dynamic network attack data. The proposed algorithm has demonstrated superior perception ability in the computer network security situation and against different types of network attacks. In practical situations, the research can provide timely and accurate situational awareness information for network administrators, helping them make quick decisions and reduce losses caused by network attacks. Meanwhile, the real-time performance of the model needs to be further optimized to better adapt to rapidly changing network security threats.

Povzetek: Študija predstavi izboljšani hibridni model v omrežni varnosti, ki dosega visoko natančnost zaznave napadov.

1 Introduction

Computer Network Security Situation Awareness (NSSA) refers to predicting network security by interpreting the potential significance of a series of security elements in a computer network. The common NSSA techniques mainly include data feature extraction, data fusion, and data visualization. Network security regulators can predict possible future network security issues based on the current network security situation, and take proactive measures to defend in advance [1-2]. Network attacks, as a key factor in computer network security incidents, have a huge impact on people's lives and economy due to their diversity. For example, worms can block communication and harm hosts, while DOS attacks can directly cause communication paralysis [3-4]. Fuzzy neural network, as an emerging intelligent computing method, combines the reasoning of fuzzy logic with the self-learning ability of neural networks, making its advantages in the field of network security increasingly significant. Fuzzy neural networks can effectively deal with the vagueness and uncertainty of data by converting the input data into a fuzzy set through fuzzification and reasoning with fuzzy rules. Compared with traditional machine learning methods, fuzzy neural networks perform better when dealing with data with fuzzy boundaries, which can identify complex network attack behaviors more flexibly. At present, scholars have carried out research on computer NSSA. The summary of relevant work is shown in Table 1.

Table 1: Related works summary table

References	Author/year	Method used	Dataset	Numerical result
Reference 5	Xiuli Du / 2023	Grey Wolf Optimization (GWO) algorithm; Clockwork Recurrent Neural Network (CW-RNN)	UNSW-NB15	The accuracy is 0.92000, the precision is 0.90575, the recall is 0.91000, and the F1-score is 0.90786
Reference 6	Adel Alqudhaibi/2023	Quadratic SVM	Self-collected dataset	The accuracy is 64.8%, the precision is 57.0%, the recall is 91.0%, and the F1-score is 70.09%.
Reference 7	Surjeet Dalal/2023	QUEST model	MSCAD	The accuracy during the training phase is 94.09%, and that during the testing phase is 94%.
Reference 8	Hongwu Zhang / 2023	Hierarchical model, grey relational analysis	Self-collected dataset	The accuracy is as high as 98%.

Du X et al. designed a clock recurrent neural network to optimize the accuracy and real-time prediction of network security status. The grey wolf algorithm was introduced to optimize the network parameters. The results showed that the model extracted features of network security status better than that of other network models, and the prediction accuracy was also improved [5]. Alqudhaibi et al. designed a prediction approach based on machine learning for critical infrastructure network attacks to address security risks and vulnerabilities in industrial control systems. The accuracy of the model reached 96% [6]. Dalal S proposed a highly boosting neural network to detect multi-stage network attacks in response to the increasing number of network attacks on firewalls. The Bayesian network model had a prediction accuracy of 97.29% for multi-stage network attacks, while the extremely enhanced neural network had an accuracy of 99.72% for multi-stage network attacks [7]. Zhang H et al. improved the accuracy of hierarchical NSSA data fusion in cloud computing environments. A hierarchical model was built to obtain a hierarchical structure for data fusion. The data fusion accuracy reached 98%, which improved the NSSA ability [8]. Although the methods proposed in the above research can make high-level predictions on the network security situation, they have not optimized the relevant algorithms, and the research object is only targeted at a single type of network attack. Therefore, a computer NSSA model based on Particle Swarm Optimization-Error Back Propagation Algorithm (PSO-BP) is designed. The research aims to improve the detection accuracy and response speed of different types of network attacks by optimizing the NSSA model, thereby effectively reducing the losses caused by network attacks and enhancing the overall security of network systems. The innovation of the research lies in optimizing gradient descent by introducing a momentum factor, accelerating network convergence, and reducing oscillations. Meanwhile, a compression factor is introduced to optimize the PSO algorithm, improving the global search ability and convergence speed. In addition, the model combines online learning mechanisms and transfer learning strategies, enabling it to adapt to new attack types in real-time, quickly adjust parameters, and enhance its ability to identify new attacks. Finally, the model introduces a hybrid update strategy. Specifically, during each step of parameter update, the model randomly selects the PSO or BP algorithm for update. This hybrid update mechanism effectively balances the global search and local search capabilities, avoiding the traditional BP algorithm from easily falling into local optima. Compared with adaptive fuzzy control, the proposed model not only combines the advantages of fuzzy logic and neural networks, but also optimizes through PSO and BP algorithms, making it particularly suitable for processing data with fuzzy boundaries. In addition, the momentum mechanism and compression factor further improve the optimization efficiency, making it perform well in handling complex network attack data. In contrast, adaptive fuzzy control mainly achieves complex system control through fuzzy logic systems, with a focus on generating and adjusting fuzzy rules. The research aims to optimize the predictive ability of computer network security situation and reduce the losses caused by network attacks. The research can provide theoretical reference for developing real-time network security intervention measures.

2 Methods and materials

2.1 Construction of computer network situation awareness model

The computer NSSA refers to the set of characteristics that can affect the network situation, such as various security threats, vulnerabilities, etc. that exist in the computer network. The common security situational awareness models include Boyd control loop model, OODA model, JDL model, etc [9-10]. To optimize the NSSA, based on traditional perception models, optimization measures are made. The overall computer NSSA model is displayed in Figure 1.

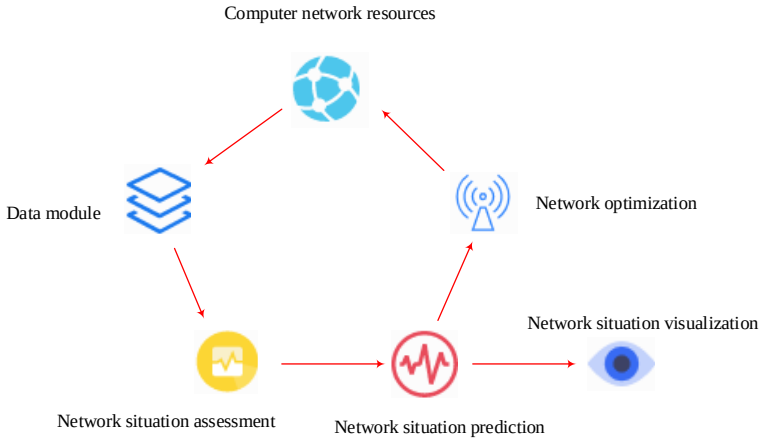


Figure 1: Diagram of network situational awareness model

In Figure 1, the computer NSSA mainly has the following four parts. Firstly, the data module is the foundation of the integrated perception model. After obtaining network security situation data from different sources, the most typical data features are selected. Finally, the data is preprocessed by removing outliers and filling in missing values [11-12]. After obtaining relevant data modules, the network security situation is evaluated based on the preprocessed data mentioned above. Fuzzy evaluation method and Principal Component Analysis (PCA) are introduced to obtain situation values, which reflect the current state of network operation. The situational value is calculated based on the weights of several features represented by PCA. Then, the data is classified and evaluated using fuzzy evaluation method. The PCA method first standardizes the data by defining an initial matrix X, which is mathematically expressed as formula (1).

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{q1} & x_{q2} & \cdots & x_{qm} \end{pmatrix} \tag{1}$$

In formula (1), m represents the data dimension. q represents the amount of data. Firstly, the initial matrix is normalized, as shown in formula (2).

$$\overline{x}_{jk} = \frac{x_{jk} - E_j}{\sigma_j} \tag{2}$$

In formula (2), x_{jk} represents the j-dimensional feature of the k-th data. σ_j signifies the variance of the j-th dimension. E_j signifies the mean of the j-th dimension. After normalization, the matrix is transposed to obtain X^T. The correlation coefficient matrix is calculated based on this transposed matrix. The relevant mathematical expression is shown in formula (3).

$$\begin{cases} CCM = (ccm_{jk}) \\ ccm_{jk} = (\sigma_{jk}) / \sqrt{\sigma_{kk}} \square \sqrt{\sigma_{jj}} \end{cases} \tag{3}$$

In formula (3), ccm_{jk} represents the correlation coefficient. σ_{jk} signifies the variance corresponding to the k-th feature of the j-th data. σ_{kk} represents self-variance. σ_{jj} signifies the variance of the j-th data. The corresponding eigenvalues and eigenvectors are obtained through CMM. The variance contribution rate is calculated, as displayed in formula (4).

$$\alpha_k = \lambda_k / (\sum_{k=1}^{m} \lambda_k) \tag{4}$$

In formula (4), α_k represents the variance contribution rate corresponding to each principal component. λ represents the characteristic root. The variances of the first d principal components are summed. If the value is greater than 0.95, the resulting d principal components are defined as the principal components after dimensionality reduction. Therefore, the mathematical expression of principal components based on dimensionality reduction is shown in formula (5).

$$F_v = L_{1v}X_1 + L_{2v}X_2 + L_{3v}X_3 + \cdots + L_{mv}X_m \quad v \prec d \tag{5}$$

In formula (5), L represents the combination coefficient of the initial matrix. Based on formula (5) and combined with variance contribution rate, the situation value of computer network security is between $[0,1]$. After normalization, the mathematical expression of the weights is finally obtained, as shown in formula (6).

$$\begin{cases} w_k = \psi_k / (\sum_{k=1}^{m} \psi_k) \\ \psi_k = (\sum_{v=1}^{d} L_{kv}\alpha_1) / (\sum_{k=1}^{d} \alpha_1) \end{cases} \tag{6}$$

In formula (6), w_k represents the final weight. To evaluate the situation value, the research introduces the fuzzy evaluation method to assess the types of data features. The number of fuzzy subsets is $b = \{b_1, b_2, b_3, \cdots b_o\}$. The membership function $\mu_{ab}(a = 1,2,3,\cdots m; b = 1,2,3,\cdots o)$ for each fuzzy subset is defined. Among them, μ_{ab} is the membership degree of the a-th indicator in the b-th domain. The research first introduces the K-means Clustering (K-means) based on membership functions, and uses the Calinski-Harabaz index to characterize the number of clusters. The index is the ratio of inter group dispersion to intra group dispersion. The larger the value, the better the clustering effect [13-14]. The relevant mathematical expression is shown in formula (7).

$$\begin{cases} s = \dfrac{tr(B_f)}{tr(W_f)} \square \dfrac{n_E - f}{f - 1} \\ W_f = \sum_{t=1}^{f} \sum_{x \in C_t} (x - c_q)(x - c_q)^T \\ B_f = \sum_{t=1}^{f} p_q (c_q - c_E)(c_q - c_E)^T \end{cases} \tag{7}$$

In formula (7), W_f represents the discrete matrix within the group. B_f represents the inter group discrete matrix. C_q signifies the point set of clusters q. c_q signifies the center of cluster q. p_q is the number of points in cluster q. n_E represents the sample size. f

indicates the number of clusters. After specifying the number of clusters, f samples are defined as cluster center points. The cluster center point corresponding to the first sample is taken as the initial value. The remaining samples are adjusted to the class with the smallest distance using Euclidean distance. Finally, the average of each group is calculated, and the center of the group is updated. After determining the number of clusters and cluster centers, the research selects Gaussian functions as membership functions. The mean of the Gaussian function corresponds to the center vector c discussed in the above, and the variance is calculated in formula (8).

$$\sigma_i = c_{\max} / \sqrt{2f} \tag{8}$$

In formula (8), σ_i represents the variance of the membership function. $c_{\max}$ represents the maximum distance. Combining the center vector and variance, the mathematical expression of the membership function is shown in formula (9).

$$\mu_{ij} = \exp\left\{\frac{-(x_i - c_{ij})^2}{\sigma_i^2}\right\} \tag{9}$$

In formula (9), μ_{ij} represents the membership function of the i-th feature based on the j-th class. After determining the membership function, the membership degree of a single feature is counted. Based on the corresponding membership relationship, the related fuzzy relationship matrix is shown in formula (10).

$$N = \begin{pmatrix} n_{11} & n_{12} & \cdots & n_{1f} \\ n_{21} & n_{22} & \cdots & n_{2f} \\ \vdots & \vdots & \vdots & \vdots \\ n_{m1} & n_{m2} & \cdots & n_{mf} \end{pmatrix} \tag{10}$$

In formula (10), $n_{mf} = \mu_{mf}(x_m)$ can be obtained through the corresponding relationship. Based on the matrix N, fuzzy relationship calculation is performed. The term with the largest single feature is selected as the output. Therefore, for the maximum value of each row of the matrix, the matrix FN is obtained through transposition. Combined with the weight values discussed in the above research, the final normalization is shown in formula (11).

$$\begin{cases} \overline{FER} = \dfrac{FER}{\sum FER} \\ FER = \sum_{k=1}^{m} fn_m w_m \end{cases} \tag{11}$$

According to formula (11), the final fuzzy evaluation result can be obtained. The computer network situational awareness model mainly consists of an input layer, a member layer, a rule layer, a normalization layer, a fuzzy logic layer, and an output layer. The specific input layer has a total of 43 units, the affiliation layer has 43×3 units, the rule layer, the normalization layer, and the fuzzy logic layer each have 100 units, and the output layer has 1 unit, representing the probability of attack occurrence. In terms of parameters, the weight initialization range is [-0.1, 0.1], and a random initialization method is adopted to ensure that the model has good convergence in the early stage of training. The centers and variances of the membership function are determined by the K-means clustering algorithm. The cluster centers serve as the mean of the membership function, and the variances are calculated by formula (8). The evaluation metrics used are Accuracy, Recall, F1-score, False Positive Rate (FPR), False Negative Rate (FNR), False Alarm Rate (FAR). The threshold for attack detection is set to 0.5, that is, when the output probability is greater than 0.5, it is judged as an attack. When it is less than or equal to 0.5, it is judged as a normal flow rate. For each performance indicator, the research sets the confidence interval at 95%. Based on the research, the uncertainty and variability of the model performance can be more comprehensively demonstrated.

2.2 Optimization of computer network situation awareness model based on fuzzy neural network algorithm

Based on the computer NSSA model in the above research, the research first analyzes the structure of the fuzzy neural network. Based on these advantages of the fuzzy neural network, an improved T-S fuzzy neural network structure is proposed. The standard network structure is relatively simple and can be widely applied in simple fuzzy reasoning. The T-S fuzzy neural network structure is more complex compared with the standard type. The structure is mainly divided into a antecedent network and a posterior network. The former has an input layer, a membership layer, a rule layer, and a normalization layer, while the latter has a fuzzy logic layer and an output layer that are separately owned by the T-S network [15-16]. An improved T-S type network structure is proposed, as presented in Figure 2.

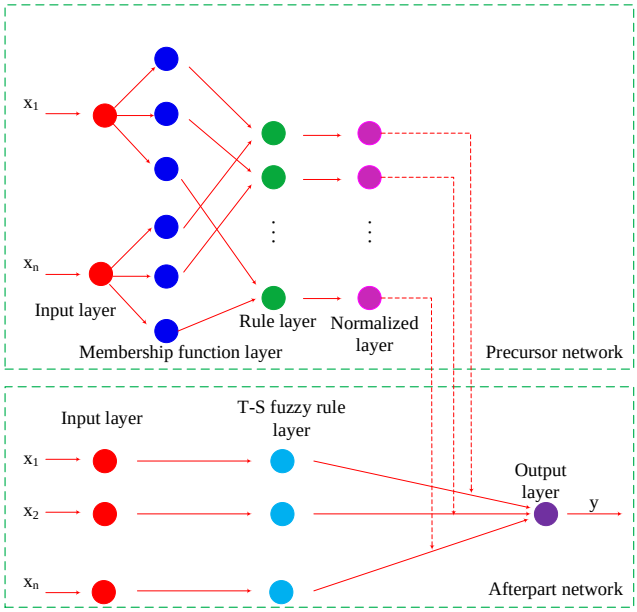


Figure 2: Improved T-S neural network structure diagram

In Figure 2, for the antecedent network, the first layer is the input layer, which inputs the features of the data into the neural network model. The membership function layer uses the Gaussian function proposed in the above research as the membership function to fuzzify the input features, and combines all features into a fuzzy set. The fuzzy subset output by the membership function layer is used as input for fuzzy rule output. The output of the rule layer is shown in formula (12).

$$\xi_j = \mu_1^n \mu_2^n \mu_2^n \cdots \mu_m^n \quad n = 1,2,3,\cdots f \tag{12}$$

In formula (12), f represents the number of subordinate layers. Finally, by normalizing the rule layer, the normalized output value is displayed in formula (13).

$$\overline{\xi_j} = \frac{\xi_j}{\sum_{r=1}^{f} \xi_r} \tag{13}$$

In formula (13), $\overline{\xi_j}$ represents the normalized output value. For the backend network part of the fuzzy neural network, after passing through the input layer, the fuzzy logic layer performs fuzzy inference based on the data features of the input layer. The output of the fuzzy logic layer combines the normalized output values of the frontend network. Therefore, the output of the fuzzy logic layer is shown in formula (14).

$$\eta_j = \sum_{i=1}^{m} p_{ij} x_i \quad i = 1,2,3,\cdots m \quad j = 1,2,3,\cdots f \tag{14}$$

In formula (14), p_{ij} signifies the corresponding weight parameter. Finally, η_j enters the output layer. The Sigmoid activation function is introduced to optimize the generalization ability. Therefore, the final output of the fuzzy neural network is shown in formula (15).

$$\begin{cases} y = h(\eta_1 \xi_1 + \eta_2 \xi_2 + \eta_3 \xi_3 + \cdots \eta_f \xi_f) \\ h(x) = 1/(1+e^{-x}) \end{cases} \tag{15}$$

In formula (15), $h()$ represents the Sigmoid activation function. Regarding learning algorithms, the research introduces Error Back Propagation (BP) and Particle Swarm Optimization (PSO) for NSSA. The BP algorithm mainly optimizes parameters through gradient descent. Compared with the traditional BP neural network, the PSO-BP algorithm optimizes the weights and thresholds of BP neural network through the global search capability of PSO, which avoids the problem that BP algorithm is easy to fall into local optimal. The global optimization capability of PSO makes up for the problem that the traditional BP neural network is sensitive to the initial weight, and significantly improves the stability and convergence speed of the model. Considering the increasing abundance of novel cyber-attacks, the research first introduces an online learning mechanism into the model. Online learning allows the model to continuously receive new data during operation and update the model parameters in real-time. The online learning mechanism is implemented through incremental learning. During its operation, the model continuously receives new data and updates its parameters in real-time. The specific implementation steps are as follows. Firstly, the newly received data are preprocessed, including normalization, feature extraction, etc. Secondly, the model parameters with new data are updated and optimized using the PSO-BP algorithm. Finally, the model performance is evaluated in real-time to ensure that the model maintains high detection accuracy in dynamic environments. Meanwhile, for some new types of attacks that are

similar to known attack types, the research introduces a transfer learning strategy by transferring existing model knowledge to new attack scenarios. The model can quickly adapt with the support of a small amount of new data, especially when it detects new variants of attacks. It uses existing model weights as initial values and only fine tunes some parameters to improve the model's ability to recognize new attacks. The transfer learning strategy is implemented by transferring the existing model knowledge to new attack scenarios. Firstly, using the existing model weights as initial values, some parameters are fine tuned. Secondly, a small amount of new data is used to fine tune the model to ensure that it can quickly adapt to new attack scenarios. Finally, the performance of the model in the new attack scenario is verified through experiments to ensure the adaptability and generalization ability of the model. In the above research, the output of the neural network model is y. Combined with the expected true value of the output, the squared loss function of the neural network is presented in formula (16).

$$L = \frac{1}{2}(y^c - y)^2 \tag{16}$$

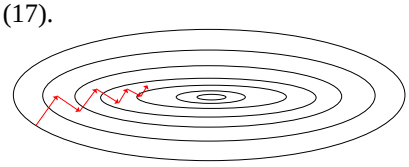
In formula (16), y^c signifies the expected true value of the output. L signifies the loss function. Based on formula (16), when the expected value is close to the actual value, a smaller L value indicates better fitting performance. The optimization of the loss function in the BP is mainly in the backward propagation stage, so gradients are used in the propagation process to update weights, membership function centers, and variances, as shown in formula (17).

$$\begin{cases} p_{ij}(t+1) = p_{ij}(t) - \alpha \dfrac{\partial L}{\partial p_{ij}} \\ c_{ij}(t+1) = c_{ij}(t) - \alpha \dfrac{\partial L}{\partial c_{ij}} \\ (t+1) = \sigma_i(t) - \alpha \dfrac{\partial L}{\partial \sigma_i} \end{cases} \tag{17}$$

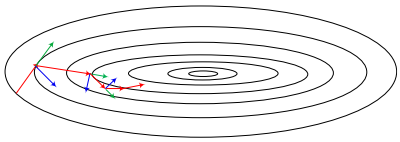
In formula (17), $p()$ represents the weight. $c()$ represents the center of the membership function. $\sigma()$ represents the variance. When using gradient descent for parameter updates, the direction of single optimization is perpendicular to the direction of the previous update. During this process, there is often slow convergence speed in the network. Therefore, the research introduces momentum to optimize the optimization direction of gradient descent parameter updates. By introducing the momentum factor, the convergence of the network can be accelerated and the oscillations during the gradient descent can be reduced. The related mathematical expression is shown in formula (18).

$$v(t+1) = \mu v(t) + \eta \nabla E(t) \tag{18}$$

In formula (18), $v(t)$ denotes the velocity at the current moment. μ denotes the momentum factor. η denotes the learning rate. $\nabla E(t)$ denotes the current gradient. The momentum factor optimizes the direction and speed of parameter update by combining the velocity of the previous moment and the direction of the current gradient, and the related schematic is shown in Figure 3.



(a) Initial BP algorithm gradient descent diagram



(b) Gradient descent diagram of BP algorithm with momentum

Figure 3: Schematic diagram of introducing momentum gradient descent to find the optimal directional contrast

Figure 3 (a) shows the optimization direction of gradient update in the original state, which is perpendicular to the previous direction. Figure 3 (b) shows the optimization direction after introducing momentum. The red and blue lines represent the gradient direction, the green line signifies the velocity direction, and the red line signifies the optimized optimization direction based on the vector operation criterion. Therefore, the main idea is to use the direction updated last time as the basis, and combine it with the gradient as the update direction, which can make the network structure faster. The correction of relevant parameters is shown in formula (19).

$$\begin{cases} v_t = d\Phi_t + qv_{t-1} \\ \eta_{t+1} = \eta_t - v_t \end{cases} \tag{19}$$

In formula (19), v_t and v_{t-1} represent the current and previous updates, respectively. d represents the efficiency of gradient learning. Φ_t represents the gradient. q represents the momentum factor. The PSO algorithm is further used to learn parameters. The research makes the following improvements based on the basic PSO. A compression factor is applied to accelerate the convergence speed of the PSO and optimize the computational efficiency. The compression factor is introduced, which can dynamically adjust the speed of the particles to avoid premature convergence of the particles to the local optimal solution. The mathematical expression based on the compression factor is shown in formula (20).

$$\begin{cases} \lambda = \dfrac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|} \\ \varphi = c_1 + c_2 \end{cases} \tag{20}$$

In formula (20), λ represents the compression factor. c_1 and c_2 represent learning factors. In the PSO algorithm, the momentum and compression factors can significantly improve the convergence speed and global search ability of the algorithm. Specifically, first, the position k and velocity k of the particles are initialized, while the individual optimal position k and the global optimal position k are initialized. Then, the position of the particle is updated based on formula (21).

$$\begin{aligned} & v_i(t+1) = q \cdot v_i(t-1) + \lambda \cdot (\omega \cdot v_i(t)) \\ & + c_1 \cdot r_1 \cdot (gbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t))) \cdot x_i(t) \\ & = x_i(t) + v_t(t+1) \end{aligned} \tag{21}$$

After iterative updates based on formula (21), the fitness of each particle is evaluated, and the individual optimal position and the global optimal position are updated. The iteration stops when the maximum number of iterations is reached or the convergence conditions are met. Based on the optimized BP and PSO discussed in the above research, a new PSO-BP is formed by combining the two to complete the parameter learning. The steps based on the PSO-BP are as follows. Firstly, the parameters of PSO and BP are initialized, and the selection probability p is defined. A number between [0,1] is randomly generated and compared it with the selection probability. If the random number is less than the random probability, PSO is applied to update the parameters. Otherwise, BP is applied to update the parameter values. Whether the maximum iteration is reached or the optimal value has been found is determined. If it meets the requirement, it ends. Otherwise, it continues to generate random numbers and repeats the above steps. Finally, the pseudo-code of PSO-BP is shown in Table 2.

Table 2: PSO-BP pseudo-code

```
Algorithm: PSO-BP
Input: Training dataset D, number of particles p_size, maximum iterations max_iter
Output: Optimized neural network weights W

Initialize:
    Initialize the neural network weights W randomly
    Initialize the particle swarm with p_size particles
    For each particle i:
        Initialize the position x_i and velocity v_i randomly
        Set the personal best position pbest_i to x_i
        Evaluate the fitness of x_i using the training dataset D
    Set the global best position gbest to the best pbest_i

Main Loop:
    For iteration = 1 to max_iter:
        For each particle i:
            Update the velocity v_i using:
                v_i = w * v_i + c1 * rand() * (pbest_i - x_i) + c2 * rand() * (gbest - x_i)
            Update the position x_i using:
                x_i = x_i + v_i
            Evaluate the fitness of x_i using the training dataset D
            If the fitness of x_i is better than pbest_i:
                Update pbest_i to x_i
            If the fitness of x_i is better than gbest:
                Update gbest to x_i
        Update the neural network weights W using the global best position gbest
        Perform backpropagation to fine-tune the weights W
        If the stopping criterion is met (e.g., convergence or max_iter reached):
            Break

Return the optimized neural network weights W
```

3 Results

3.1 Performance analysis based on PSO-BP

The experimental environment parameters and PSO-BP algorithm parameter settings are displayed in Table 3.

Table 3: Experimental environment parameters

Parameter name	Parameter model	Parameter name	Parameter model
CPU	Intel® Core™ i5-12500H	Inertial weighting	0.5
RAM	64GB	c1	2
GPU	NVIDIA GeForce RTX 3050	c2	2
Kernel	14 nuclei	BP algorithm learning rate	0.1
Display memory	4GB	Gradient learning rate	
Development language	Python3.8	Random numbera a	0.5
Related libraries	Keras 、 Slearn	Maximum iterations	100
Number of particles p_size	50	Center Particle range	[0,1]
Delta particle range	[0,0.2]	P particle value range	[-50,50]
PSO algorithm learning rate	0.1	Penalty factor	0.3

Batch size	32	Epochs	50

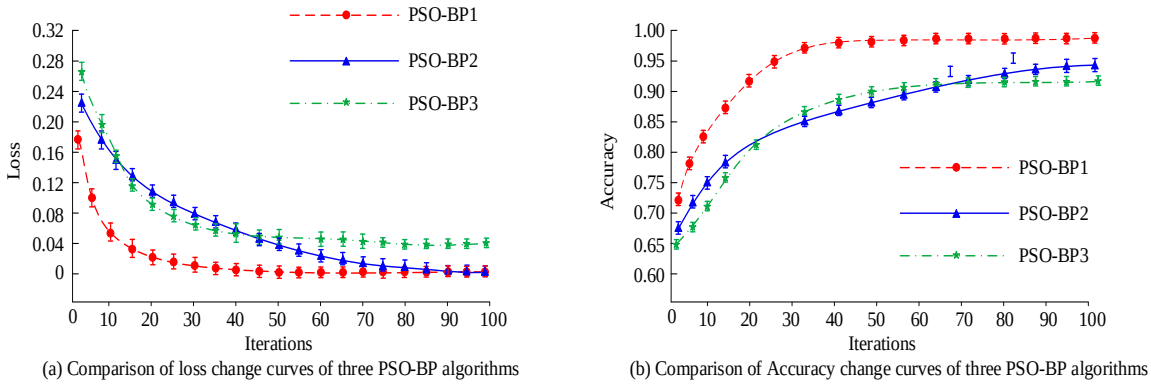
After setting the parameters in Table 3, the UNSW-NB15 dataset is selected to assess the performance of the PSO-BP. The dataset is a large dataset developed by the University of New South Wales in Australia for research on network intrusion detection systems, capable of simulating normal and abnormal network traffic in real network environments. Subsequently, the data in the dataset is preprocessed, dividing into training and testing sets in a 6:4. The data set is randomly divided into K subsets, ensuring that each subset is equal in size. In each iteration, a subset is selected as the validation set, and the remaining K-1 subsets are merged into the training set. It is repeated K times, ensuring that each subset is used as a validation set once. Meanwhile, SMOTE oversampling is adopted. By inserting new synthetic samples between the minority class samples, the number of minority class samples is increased to balance the dataset. Simulation experiments are performed on six different types of network attack types in the dataset, including DoS, Fuzzers, and Shellcode. The sample distribution and attack types of the dataset are displayed in Table 4.

Table 4: Sample distribution and attack types in the dataset

Attack type number	Training set	Testing set
A1	220	147
A2	4,333	2,888
A3	28,456	18,970
A4	5,098	3,398
A5	754	503
A6	109	72

After completing the classification of attack types in the dataset. The performance of the research method is assessed using the training set. The index includes loss function, accuracy, recall, and F1-score. The PSO-BP algorithm discussed in references [17] and [18] is selected for comparative experiments [17-18]. In each iteration of PSO-BP1, a number between [0,1] is randomly generated and compared with the selection probability. If the random number is less than the selection probability, the PSO algorithm is used to update the parameters. Conversely, the parameter values are updated based on the BP algorithm. PSO-BP2 only uses the PSO algorithm for parameter update and does not combine the BP algorithm. PSO-BP3 only uses the BP algorithm for parameter update and does not combine the PSO algorithm. The loss function and accuracy variation curve on the training set are shown in Figure 4.

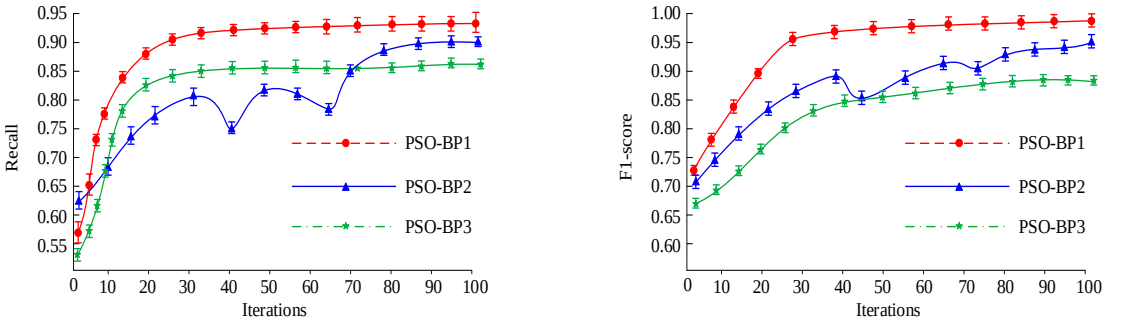
Figure 4 (a) shows the loss function curves of three PSO-BP algorithms. The loss function usually represents the difference between the output value and the true value of the network model. The loss function values of all three algorithms decreased with the increase of iterations. The research method (PSO-BP1) tended to flatten the loss function values when the iteration exceeded 30, and the loss function value eventually converged to around 0.01. The comparison algorithm PSO-BP2 had the same loss function convergence value as the PSO-BP1, but its convergence speed was slow. The loss function value only flattened at around 90 iterations. Finally, the PSO-BP3 algorithm had the maximum loss function convergence value, which was around 0.04. Figure 4 (b) displays the accuracy curves. The accuracy of all three methods increased with the increase of iterations. The accuracy of the PSO-BP1 outperformed the other two algorithms, and its accuracy eventually converged to around 0.99. The recall rate and F1-score are further compared, as displayed in Figure 5.



(a) Comparison of loss change curves of three PSO-BP algorithms

(b) Comparison of Accuracy change curves of three PSO-BP algorithms

Figure 4: Comparison of loss function and accuracy curve of three PSO-BP algorithms



(a) Comparison of Recall change curves of three PSO-BP algorithms

(b) Comparison of F1-score change curves of three PSO-BP algorithms

Figure 5: Comparison of recall and F1-score curves of three algorithms

Figure 5 (a) displays the recall curves. The recall of the PSO-BP1 and PSO-BP3 increased with the increase of iterations. However, PSO-BP2 exhibited oscillations between 30-70 iterations, which may be due to over-fitting of the model, leading to fluctuations in the recall within the iteration region. The final recall of the PSO-BP1 converged to around 0.94, which outperformed other algorithms. Figure 5 (b) shows the F1-score curve for three algorithms. The PSO-BP1 had the highest F1-score of 0.73 in the initial state, followed by PSO-BP2 and finally PSO-BP3. After convergence of three methods, the F1-score of the PSO-BP1 exceeded the other two algorithms, and the final F1-score converged to around 0.98. A K-fold cross-validation instrument is introduced to verify the performance of the module under different datasets, where the K is set to 5. Therefore, the performance indicators based on K-fold cross-validation are shown in Table 5.

In Table 5, the PSO-BP1 is superior to the other two PSO-BP models in various performance indexes. In terms of accuracy, the average accuracy of the PSO-BP1 reached 98.1%, the average recall reached 96.5%, the average F1-score was 97.3%, and the FPR, FNR, and FAR were 1.8%, 3.5% and 3.3%, respectively. Through K-fold cross-validation, the stability and generalization ability of the model under different data partition are verified. The experimental results show that the performance indexes of the PSO-BP1 fluctuate less and the standard deviation is low under different fold numbers, which indicates that the model has good universality and robustness. The research further introduces Rule-Based System (RBS), traditional BP neural network, and Support Vector Machine (SVM) for comparative analysis. The results are shown in Table 6.

Table 5: Cross-validation results

Number of folds	Algorithm name	Accuracy	Recall	F1-score	FPR	FNR	FAR
1	PSO-BP1	0.982	0.965	0.973	0.018	0.035	0.032
2	PSO-BP1	0.978	0.959	0.968	0.021	0.041	0.038
3	PSO-BP1	0.985	0.972	0.978	0.015	0.028	0.029
4	PSO-BP1	0.981	0.960	0.971	0.019	0.040	0.035
5	PSO-BP1	0.983	0.968	0.975	0.017	0.032	0.031
1	PSO-BP2	0.965	0.94	0.952	0.025	0.060	0.045
2	PSO-BP2	0.96	0.935	0.947	0.028	0.065	0.05
3	PSO-BP2	0.968	0.945	0.956	0.023	0.055	0.048
4	PSO-BP2	0.962	0.938	0.95	0.026	0.062	0.052
5	PSO-BP2	0.967	0.942	0.954	0.024	0.058	0.049
1	PSO-BP3	0.958	0.93	0.943	0.030	0.070	0.055
2	PSO-BP3	0.955	0.925	0.939	0.032	0.075	0.060
3	PSO-BP3	0.96	0.932	0.946	0.029	0.068	0.057
4	PSO-BP3	0.957	0.928	0.942	0.031	0.072	0.059
5	PSO-BP3	0.962	0.935	0.948	0.028	0.065	0.054

Table 6: Performance analysis of PSO-BP algorithm and control algorithm

Algorithm name	PSO-BP1	RBS	BP	SVM
Accuracy	0.973	0.885	0.924	0.946
Recall	0.941	0.852	0.897	0.913
F1-score	0.982	0.876	0.919	0.938
FPR	0.02	0.063	0.048	0.039
FNR	0.035	0.102	0.076	0.068
FAR	0.036	0.095	0.062	0.054
Training time	128.4s	20.3s	150.6s	142.7s

In Table 6, the accuracy, recall, and F1-score of the PSO-BP1 were higher than those of the other three algorithms, indicating its high accuracy in detecting network attacks. Its FPR, FNR, and FAR were lower than those of other algorithms, indicating its excellent performance in reducing misjudgment. Finally, the training time was only 128.4s, which was moderate and suitable for real-time network security situation awareness. The accuracy and recall of RBS were low, and the FPR and FNR were high, indicating that RBS relies on predefined rules and is difficult to adapt to new attacks. The training time was short, but the flexibility was poor. The research compares the resource utilization rates of three algorithms, as shown in Table 7.

Table 7: Analysis of resource utilization of three algorithms

Algorithm name	PSO-BP1	PSO-BP2	PSO-BP3
CPU usage (%)	44.7	50.3	47.9
Memory usage (MB)	256.3	280.7	265.4
Processing time (s)	0.22	0.31	0.28
R^2	0.982	0.965	0.958
MSE	0.0031	0.0052	0.0063
RMSE	0.0548	0.0721	0.0792
MAPE	0.0117	0.0149	0.0167
MAD	0.0347	0.0413	0.0446

Table 7 shows the resource utilization comparison of the three algorithms. The CPU usage of PSO-BP1 was 44.7%, which was lower than that of PSO-BP2 (50.3%) and PSO-BP3 (47.9%), indicating that it puts less pressure on system resources. The memory usage of PSO-BP1 was 256.3MB, which was lower than that of PSO-BP2 (280.7 MB) and PSO-BP3 (265.4 MB), indicating that it is more efficient in memory utilization. The processing time of PSO-BP1 was 0.22s, which was lower than that of PSO-BP2 (0.31s) and PSO-BP (0.28s), indicating that PSO-BP1 is more efficient in single data processing. Meanwhile, the PSO-BP1 has better prediction performance index values than those of the control algorithm, indicating that its prediction accuracy as well as error are better than those of the control algorithm. The improvement of model performance by online learning mechanism, incremental learning mechanism, and transfer learning mechanism is verified through ablation experiments. The results of the ablation experiment are shown in Table 8.

Table 8: Results of the ablation experiment

Experimental setup	Accuracy	Recall	FPR	FNR
Baseline model (without dynamic learning mechanism	0.887	0.858	0.053	0.103
Online learning mechanism	0.918	0.894	0.042	0.083
Incremental learning mechanism	0.929	0.902	0.035	0.071
Transfer learning mechanism	0.915	0.895	0.045	0.085
Online learning + incremental learning	0.934	0.915	0.033	0.065
Online learning + transfer learning	0.925	0.905	0.035	0.075
Incremental learning + transfer learning	0.935	0.928	0.030	0.062
Complete model	0.945	0.936	0.025	0.052

In Table 8, the complete model demonstrated significant performance advantages. Specifically, the accuracy of the complete model reached 0.945, which was a significant improvement compared to the baseline model. This indicates that the model is more precise in identifying network attacks and normal traffic. The recall also reached 0.936, meaning that the model can detect

actual attack events more effectively and reduce the possibility of missed reports. Compared with the baseline model of 0.050 and 0.100, the FPR and FNR were respectively reduced to 0.025 and 0.052, demonstrating the outstanding ability of the complete model in reducing false positives. It can more accurately distinguish attacks from normal behaviors, avoid unnecessary alerts, and ignore real threats. These improvements not only enhance the model's detection capabilities, but also increase its adaptability and stability in dynamic network environments, enabling it to better cope with emerging attack types and constantly changing network traffic. The research further verifies the influence of the momentum factor and the compression factor on the model performance through ablation experiments. The results are shown in Table 9.

Table 9: Ablation study results

Experimental Setup	Accuracy	Recall	FPR	FNR	FAR
Baseline (No Momentum, No Compression)	0.887	0.858	0.053	0.103	0.05
With Momentum	0.918	0.894	0.042	0.083	0.045
With Compression	0.929	0.902	0.035	0.071	0.04
With Momentum and Compression	0.945	0.936	0.025	0.052	0.035

In Table 9, the performance indicators of the baseline model were relatively low, with an accuracy of 0.887, a recall of 0.858, an FPR of 0.053, an FNR of 0.103, and a FAR of 0.050. When the momentum factor and the compression factor were introduced, the performance of the model was improved. After introducing both the momentum factor and the compression factor simultaneously, the performance of the complete model was the best, with its accuracy increasing to 0.945 and the recall increasing to 0.936. Finally, the performance indicators of PSO-BP and adaptive fuzzy systems, reinforcement learning, CNN, LSTM, and Transformer models are compared in different dynamic scenarios. The results are shown in Figure 6.

Figure 6: Comparison of model performance indicators in dynamic scenarios

In Figure 6, the two-color bands of the model respectively represent dynamic scenario 1 (where new attack types gradually emerge) and dynamic scenario 2 (where attack features change dynamically). The results show that the PSO-BP model demonstrates significant advantages. In scenario 1 where new attack types gradually emerged, the accuracy of the PSO-BP model reached 0.942, the recall was 0.925, the F1-score was

0.933, and the FPR and FNR were 0.024 and 0.052, respectively. This indicates that the PSO-BP model can efficiently identify new attack types. In scenario 2 where attack features change dynamically, the accuracy of the PSO-BP model was 0.940, the recall was 0.920, the F1-score was 0.930, and the FPR and FNR were 0.025 and 0.055, respectively. This further proves the excellent adaptability and stability of the PSO-BP model in attack features. In conclusion, the PSO-BP model significantly outperforms other models in dynamic attack scenarios, such as adaptive fuzzy systems, reinforcement learning-based methods, CNN, LSTM, and Transformer, fully demonstrating its superior performance in NSSA.

3.2 Analysis of network security situation awareness based on PSO-BP algorithm

In the above research, it is explored that the proposed PSO-BP has better performance indicators on the testing set. Furthermore, the PSO-BP is characterized on the testing set to evaluate its perception performance for six types of attacks. In addition to accuracy, the research introduces FPR, FNR, and FAR. The accuracy of network attack situation recognition based on the three algorithms is shown in Figure 7.

In Figure 7, the recognition accuracy of the proposed PSO-BP for all attack types was significantly higher than that of the other algorithms. The accuracy of A6 was as high as 98.9%, which represents an attack type like worms, referring to the self-replicating malicious software that destroys the system by occupying network bandwidth. Worm network attacks mainly spread by scanning for vulnerabilities in the network, such as operating system vulnerabilities and software vulnerabilities. After infecting the target device, they automatically replicate and continue to search for new targets. The type of network attack with the lowest recognition rate was A2, which stands for denial-of-service attack. It refers to the attack that limits the availability of the service by overloading the target server or network, ultimately making it unusable. In addition, DoS attacks may mask attack characteristics through distributed attack sources. The first mode of DOS attack is SYN Flood, that is, sending a large number of ping command packets to the target host to consume the resources of the target host. Specifically, the statistical analysis results of the accuracy for different attack types are shown in Table 10.

Table 10: Statistical analysis results of model accuracy

Attack type	PSO-BP 1	PSO-BP 2	PSO-BP 3	*p* (PSO-BP1 vs PSO-BP2)	*p* (PSO-BP1 vs PSO-BP3)
A1	0.988	0.95	0.955	0.023	0.031
A2	0.956	0.92	0.925	0.045	0.038
A3	0.986	0.955	0.96	0.018	0.027
A4	0.959	0.93	0.935	0.029	0.039
A5	0.958	0.932	0.938	0.033	0.041
A6	0.989	0.965	0.97	0.012	0.02
Average accuracy	0.973	0.942	0.948	0.025	0.035

In Table 10, the accuracy of the PSO-BP1 for different types of attacks was 98.8%, 95.6%, 98.6%, 95.9%, 95.8%, and 98.9%, respectively. Its average accuracy was 97.3%, which was an average improvement of 8.8% and 4.7% compared with the other two algorithms. Meanwhile, on all attack types, the *p* values were all less than 0.05, indicating that the accuracy of PSO-BP1 against different attack types was significantly better than that of PSO-BP2 and PSO-BP3. The FPR and FNR values of three algorithms for different types of attacks are shown in Figure 8.

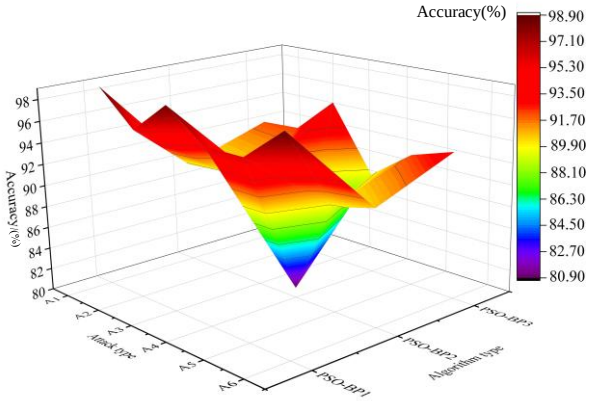
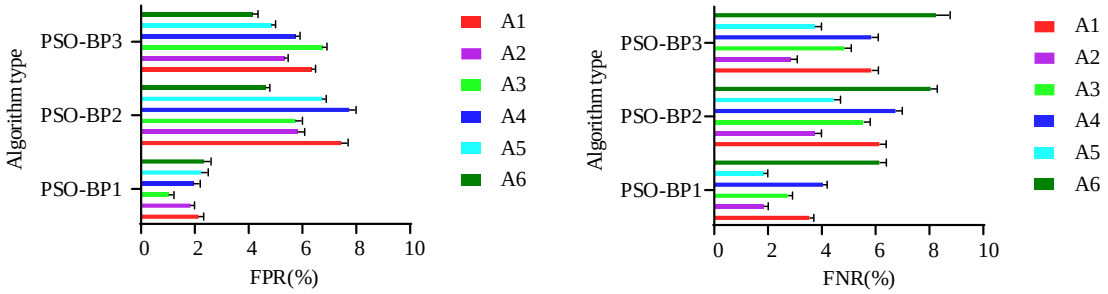


Figure 7: The accuracy of three algorithms based on various attack types



(a) Comparison of FPR values of three algorithms

(b) Comparison of FNR values of three algorithms

Figure 8: Comparison of FPR and FNR values of three algorithms based on different attack types

In Figure 8 (a), the overall FPR value of the PSO-BP1 was the lowest, and the FPR value for A3 was also the lowest. This represents vulnerability exploitation, mainly referring to conducting cyber attacks by taking advantage of software vulnerabilities. The FPR values of the PSO-BP1 were 2.1%, 1.9%, 1.1%, 2.0%, 2.3% and 2.4% respectively. The average FPR value was 2.0%, which was on average 4.5% and 3.5% lower than that of the other two algorithms. In Figure 8 (b), the PSO-BP1 has the lowest FNR value, among which A5 had the lowest FPR value, which represents the vulnerability code. By exploiting the vulnerability in the system, it enables intruders to run arbitrary code on the system. The FNR values of the PSO-BP1 were 4.1%, 1.9%, 2.8%, 4.1%, 1.9% and 6.2%, respectively, with an average FNR value of 3.5%. The FAR values of the three algorithms against different attack types are shown in Figure 10.

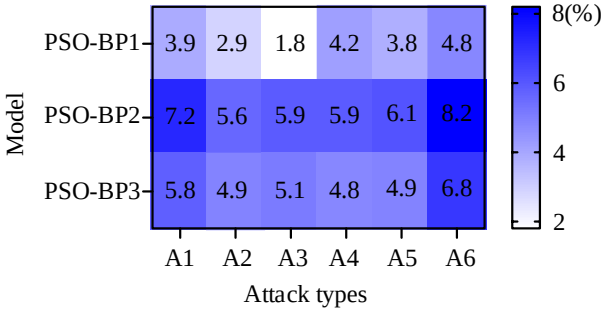


Figure 9: FAR values of three algorithms based on different attack types

Figure 9 compares the FAR values of three algorithms based on different types of network attacks. The PSO-BP1 had significantly lower FAR values than the other two algorithms in all attack types, with A3 having the lowest FAR value. Meanwhile, the PSO-BP2 algorithm had the highest FAR value for all types of network attacks, followed by PSO-BP3. The algorithm had FAR values of 3.9%, 2.9%, 1.8%, 4.2%, 3.8%, and 4.8 for all types of attacks, with an average FAR value of 3.6%, which was 2.9% and 1.8% lower than the average FAR values of the other two algorithms. The processing time, resource occupation, detection accuracy, and computational cost of the PSO-BP1 model in large-scale network environments are compared with those of the intelligent adaptive controller and the fuzzy-neural synchronization method. The results are shown in Table 11.

Table 11: Comparison results of performance indicators

Model	Processing time (s)	CPU usage (%)	Memory usage (MB)	Computational cost (s)
PSO-BP1	0.21	45.5	260.3	130.6
Intelligent adaptive controller	0.32	55.9	302.9	152.6
Fuzzy-neural synchronization method	0.28	50.2	281.3	139.8

In a large-scale real-time network environment, the PSO-BP1 model demonstrated significant performance advantages. Specifically, the processing time of the PSO-BP1 model was 0.21 seconds, while the intelligent adaptive controller was 0.32 seconds, and the fuzzy-neural synchronization method was 0.28 seconds, indicating that the PSO-BP model could more effectively handle high-dimensional data and massive network traffic. The CPU usage of the PSO-BP1 model was 45.5% and the memory usage was 260.3MB, both of which were lower than those of the intelligent adaptive controller (CPU 55.9%, memory 302.9MB) and the fuzzy-neural synchronization method (CPU 50.2%, memory 281.3MB). It is indicated that the PSO-BP1 model is more efficient in resource utilization, which is suitable for deployment in resource-constrained environments. Furthermore, the computational cost of the PSO-BP1 model was 130.6 seconds, further demonstrating its high efficiency in the training and inference processes. The research further compares the delay indicators of the three methods, among which the real-time threshold for processing delay was 0.5 seconds, the real-time threshold for system delay was 1.0 second,

the real-time threshold for transmission delay was 0.2 seconds, and the real-time threshold for queuing delay was 0.3 seconds. The comparison of delay indicators is shown in Figure 10.

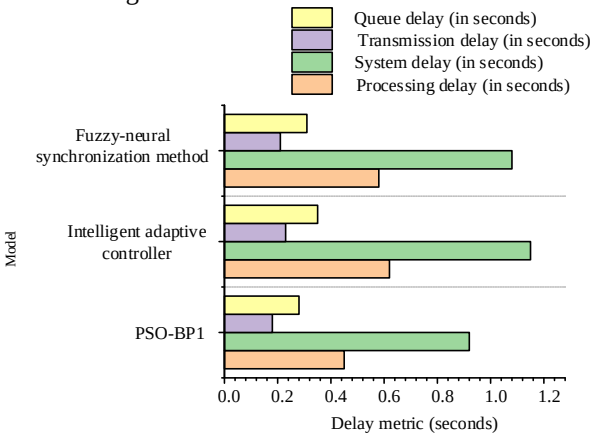


Figure 10: Comparison of delay indicators

In Figure 10, the PSO-BP1 model outperformed the other two methods in all delay indicators and was all below the real-time threshold, indicating its good performance in real-time. Specifically, its processing delay was 0.45 seconds, the system delay was 0.92 seconds, and the transmission delay and queuing delay were 0.18 seconds and 0.27 seconds, respectively. The intelligent adaptive controller has processing delay and system delay slightly higher than the real-time threshold, indicating that its performance in real-time is slightly poor. The fuzzy neural synchronization method has a system delay slightly higher than the real-time threshold, indicating that it performs moderately on real-time performance.

4 Discussion

In the research results, the loss function value of the PSO-BP1 model basically tends to level off when the number of iterations exceeds 30, and eventually converges to around 0.01. In contrast, other models such as the BP algorithm and the traditional PSO-BP algorithm have higher loss function values and slower convergence speeds under the same number of iterations. Meanwhile, the PSO-BP1 model can receive new data in real-time and update parameters through the online learning mechanism, enabling it to quickly adapt to the dynamically changing network environment. In scenarios where new attack types gradually emerge, the accuracy of the PSO-BP1 model reaches 0.942, the recall is 0.925, and the F1-score is 0.933. This indicates that the model can effectively identify new attack types while maintaining a low FPR and FNR. To evaluate the performance of the PSO-BP1 model more comprehensively, the research conducts a quantitative comparison between the PSO-BP1 and the baseline model. Specifically, the accuracy of PSO-BP1 is superior to 0.92000 of the GWO + CW-RNN model, 0.648 of the quadratic SVM model, and 0.94 of the QUEST model, but slightly lower than 0.98 of the hierarchical model + grey relational analysis. Faced with different attack types, the performance of the PSO-BP1 model in DoS is relatively lower than that of other attack types. The possible reason is that the sample size may be much larger than that of other attack types, leading to over-fitting of A2 attack type features during the training process. This data imbalance may cause the model to have deviations when distinguishing normal traffic from DoS attack traffic, thereby affecting the detection accuracy. Attackers may use camouflage techniques to hide the features of DoS attacks, making them appear as normal traffic. Currently, Mulumba D M et al. proposed a coal mine safety risk assessment model based on the PSO-BP algorithm to address certain risks in coal mining. Momentum was introduced to optimize gradient descent. Compared with other models, the neural network model had a smaller error value, with a mean square error of only 0.0002 and an average absolute percentage error of only 4.3. This research demonstrates the feasibility of the momentum-based optimization gradient descent [19]. Liu X et al. proposed a perceptual model of variational mode decomposition-PSO-BP applied to predict the permeability index. A compression factor was introduced into the particle swarm algorithm for optimization. The composite prediction model improved the accuracy by 3% compared with traditional prediction methods. The research demonstrated the feasibility of introducing a compression factor into the PSO [20]. Compared with references [19] and [20], although the PSO-BP algorithm combined with momentum and compression factor has been explored in previous literature, the research has made significant contributions in the following aspects. Firstly, a hybrid update strategy is proposed. By randomly choosing the PSO or BP algorithm for parameter update, the global search and local search capabilities have been effectively balanced, avoiding the problem that the traditional BP algorithm is prone to fall into local optimum. Secondly, an online learning mechanism and transfer learning strategy are introduced, enabling the model to adapt to new attack types in real-time and quickly adjust parameters. Finally, the data features are preprocessed and evaluated through PCA and fuzzy evaluation methods, and the weights and thresholds of the network are optimized by combining momentum and compression factors. Finally, an update mechanism can be established to enable the model to continuously learn and adapt to new attack patterns. Based on the model proposed in the research, integrating it into existing network security frameworks may enhance the overall network defense effectiveness. For example, embedding the model into existing intrusion detection systems or security information and event management systems enables them to perceive network situations and provide real-time warnings. In addition, by integrating with existing security frameworks, the model can utilize existing security data and logs to further enhance its detection capability and accuracy. In summary, the research proposes a fuzzy neural network model based

on particle swarm optimization and error back-propagation algorithm, and introduces the momentum mechanism and compression factor. This combination is highly innovative in the field of fuzzy neural networks, mainly reflected in the following aspects. Firstly, in the traditional BP algorithm, the gradient descent method is prone to getting stuck in local optimum and may experience oscillations during the convergence process. By introducing the momentum mechanism, the model can combine the velocity at the previous moment with the the current gradient direction to optimize the direction and velocity of parameter updates. Secondly, by introducing a compression factor, the model can dynamically adjust the velocity of the particles, preventing them from converging prematurely to the local optimal solution. The accuracy of the specific PSO-BP1 model reaches 0.942, the recall is 0.925, and the F1-score is 0.933. Finally, the fuzzy neural network combines the reasoning ability of fuzzy logic and the self-learning ability of neural networks, which can effectively handle the fuzziness and uncertainty of data. By introducing the momentum mechanism and compression factor into the fuzzy neural network, the adaptability and detection accuracy of the model when dealing with complex network attack data have been further enhanced.

Although the PSO-BP1 model shows certain robustness under adversarial attacks, adversarial attacks remain a serious challenge. Adversarial attacks not only can deceive the model into generating incorrect outputs, but also may lead to a decline in the model's performance on normal data. Subsequently, more advanced adversarial training methods can be explored, such as the combination of adversarial training and transfer learning, to further enhance the robustness of the model.

5 Conclusion

In the computer NSSA, the research first constructed an NSSA model, and then introduced BP algorithm and PSO algorithm. Momentum was introduced to optimize gradient descent for BP algorithm, and PSO algorithm was optimized with compression factor. Finally, they were combined to obtain PSO-BP. The relevant results showed that the proposed PSO-BP had good performance, with faster convergence speed and higher final convergence value compared with other PSO-BP algorithms. The algorithm had higher numerical accuracy and lower FAR, FPR, and FNR for different types of computer network attacks. Although significant progress has been made in the field of network security situation awareness, there are still some limitations. In the actual network environment, the types and characteristics of attacks are diverse and may occur simultaneously. A single optimization method may not be able to effectively identify and respond to all types of attacks when facing complex multi-attack scenarios, resulting in limited generalization ability and adaptability. Meanwhile, deep learning techniques can be integrated into existing PSO-BP models to further enhance their adaptability and decision-making capabilities in dynamic network environments. In addition, it can be considered to extend the PSO-BP model to the multi-agent framework to achieve distributed situation awareness. Each agent can be responsible for a sub-region in the network and improve overall situational awareness through collaboration.

References

[1] Shanmugham E K, Dhatchnamurthy S, Pakkiri P S. Adaptive activation functions with deep kronecker neural network optimized with bear smell search algorithm for preventing MANET cyber security attacks. Network: Computation in Neural Systems, 2025, 36(3): 426-450. https://doi.org/10.1080/0954898X.2024.2321391

[2] Diaba S Y, Shafie-Khah M, Elmusrati M. Cyber security in power systems using meta-heuristic and deep learning algorithms. IEEe Access, 2023, 11: 18660-18672. https://doi.org/10.1109/ACCESS.2023.3247193

[3] Badescu R. Phishing Threats in the Age of social media: A User-Centric Approach. Informatica Economica, 2024, 28(4): 83-101. https://doi.org/10.24818/issn14531305/28.4.2024.07

[4] Okeoma O, Ayoola MA, Chiedozie MO Anwuli NO, Tochukwu O, Chibuike D. Cybersecurity in U.S. and Nigeria Banking and Financial Institutions: Review and Assessing Risks and Economic Impacts. Acta Informatica Malaysia. 2023; 7(1): 54-62. https://doi.org/10.26480/aim.01.2023.54.62

[5] Du X, Ding X, Tao F. Network security situation prediction based on optimized clock-cycle recurrent neural network for sensor-enabled networks. Sensors, 2023, 23(13): 6087-6087. https://doi.org/10.3390/s23136087

[6] Alqudhaibi A, Albarrak M, Aloseel A, Jagtap S, Salonitis K. Predicting cybersecurity threats in critical infrastructure for industry 4.0: a proactive approach based on attacker motivations. Sensors, 2023, 23(9): 4539. https://doi.org/10.3390/s23094539

[7] Dalal S, Manoharan P, Lilhore U K, Seth B, Alsekait D M, Simaiya S, et al. Extremely boosted neural network for more accurate multi-stage Cyber attack prediction in cloud computing environment. Journal of Cloud Computing, 2023, 12(1): 14-26. https://doi.org/10.1186/s13677-022-00356-9

[8] Zhang H, Kang K, Bai W. Hierarchical network security situation awareness data fusion method in cloud computing environment. Journal of

computational methods in sciences and engineering, 2023, 23(1): 237-251. https://doi.org/10.3233/JCM-226542

[9] Fan Z, Zhao P, Jin B, Tang Q J, Zheng C S, Li X. Research on Key Method of Cyber Security Situation Awareness Based on ResMLP and LSTM Network. IETE Journal of Research, 2024, 70(3): 2716-2730. https://doi.org/10.1080/03772063.2023.2176365

[10] Sarjakivi P, Ihanus J, Moilanen P. Using Wargaming to Model Cyber Defense Decision-Making: Observation-Based Research in Locked Shields. European Conference on Cyber Warfare and Security. 2024, 23(1): 457-464. https://doi.org/10.34190/eccws.23.1.2270

[11] De Keersmaeker F, Cao Y, Ndonda G K, Sadre R. A survey of public IoT datasets for network security research. IEEE Communications Surveys & Tutorials, 2023, 25(3): 1808-1840. https://doi.org/10.1109/COMST.2023.3288942

[12] Guo X, Yang J, Gang Z, et al. Research on network security situation awareness and dynamic game based on deep Q learning network. Journal of Internet Technology, 2023, 24(2): 549-563. https://doi.org/10.53106/160792642023032402030

[13] Hwang Y S, Shin J. Deep Learning Assisted Intelligent Human Computer Interaction for Next Generation Internet Applications. Informatica, 2024, 48(2): 145-146. https://doi.org/10.31449/inf.v48i2.6286

[14] Saminathan K, Mulka S T R, Damodharan S. An artificial neural network autoencoder for insider cyber security threat detection. Future Internet, 2023, 15(12): 373. https://doi.org/10.3390/fi15120373

[15] Nguyen C M, Nguyen A T, Delprat S. Neural-network-based fuzzy observer with data-driven uncertainty identification for vehicle dynamics estimation under extreme driving conditions: Theory and experimental results. IEEE Transactions on Vehicular Technology, 2023, 72(7): 8686-8696. https://doi.org/10.1109/TVT.2023.3249832

[16] Gaba S, Budhiraja I, Kumar V. A systematic analysis of enhancing cyber security using deep learning for cyber physical systems. IEEE Access, 2024, 12: 6017-6035. https://doi.org/10.1109/ACCESS.2023.3349022

[17] Xu M, Zheng Y, Li Y, Wu W H. Prediction of properties of anti-breast cancer drugs based on PSO-BP neural network and PSO-SVM. Nanjing Xinxi Gongcheng Daxue Xuebao, 2023, 15(1): 51-65. https://doi.org/10.4208/JICS-2023-003

[18] Sumathi S, Rajesh R. A Dynamic BPN-MLP Neural Network DDoS Detection Model Using Hybrid Swarm Intelligent Framework. Indian Journal of Science and Technology, 2023, 16(43): 3890-3904. https://doi.org/10.17485/IJST/v16i43.1718

[19] Mulumba D M, Liu J, Hao J, Zheng Y N, Liu H Q. Application of an optimized PSO-BP neural network to the assessment and prediction of underground coal mine safety risk factors. Applied Sciences, 2023, 13(9): 5317-5317. https://doi.org/10.3390/app13095317

[20] Liu X, Zhang Y, Li X, Zhang Z F, Li H Y, Liu R, et al. Prediction for permeability index of blast furnace based on VMD-PSO-BP model. Journal of Iron and Steel Research International, 2024, 31(3): 573-583. https://doi.org/10.1007/s42243-023-01097-y